Amazon ECR 
Registry vs. Repository 
Container Registry 
Container registries store multiple repositories 
of container images, as well as storing API paths 
and access control rules just like Docker Hub. 
A repository is a place for you to publish and 
access your Docker images. It stores actual 
images. 
• You may assume that the registry consists of 
repositories and the repository contains multiple 
copies of the same image that are uniquely 
versioned with tags. 
Docker Hub and other third party repository 
hosting services are called "registries". 
Container 
Repository 
Container 
Repository 
Container 
I mage 
Container 
Image 
Container 
I mage 
Container 
Image 
Container 
I mage 
Container 
Image 

Amazon ECR container ların tutulduğu bir Amazon registry dir.

Registry repolardan oluşur. Repolar ın için de image ler bulunuyor.

Docker hub bir registry dir ancak günlük kullanımda repo da deniliyor.

Amazon ECR 
What is Amazon ECR? 
amazon 
ECR 
docker HUB 
Red Hat 
Quay 
Azure 
Container 
Registry 
HARBOR 
Amazon Elastic Container Registry (ECR) service is a container registry solution of 
AWS. 
It is a collection of repositories made to store Docker container images. 
It is also integrated with Amazon Elastic Container Service (ECS). 

AWS yi kullanan şirketler amazon ECR ı kullanabiliyor. AWS kullanan bir şirket için ECR kullanmanın avantajı AWS nin diğer servisleriyle entegre çalışmadıdır.

amazon 
ECR 
aws 
Amazon ECR Public Gallery 
ECR (Private) 

ECR ın public ve private repoları olabiliyor. Public e bir image konunca public olarak çekilebilyor. ECR ın kendi public image galerisi de var. Oradan da çekip kullanılabiliyor.

0 
Amazon ECS 
@ Amazon EKS 
AWS Cloud 
Write code 
Write and package?de 
as a Docker image 
Amazon ECR 
o 
Compress, encrypt, md 
control access to images 
o 
Version, tag. and manage 
image lifecycles 
Run containers 
Pull images and run 
containers anywhere 

Bir uygulamamız olduğunu düşünelim, onu çalıştıracak bir image oluşturduk, kubernetes cluster ya da ECR, ECS vs kullanıyor olabliiriz. Image ı build ettikten sonra Amazon ECR a gönderiyoruz. ondan sonra nerede kullanacaksak orada kullanabiliyoruz.

burada credentials devreye giriyor. role ya da secret lar (credentials tanıtıp) ile yapabiliyoruz.

ECS, EKS, Cloud ya da lokale çekebiliyoruz bu image leri.

Amazon Elastic 
Container Registry 
Private 
public reqistr,• 
Repos itori es 
Getting 
Public gall" 
@ S.fulty 
ECR > 
public 
Private repositories (3) 
c ssmp 
d e pst 
tri 
1 , amazonawscom/devopsthehardway.err.ad 
08 2023. 
IUTC•os) 
Is 2022. 
lurc•os) 
19 Ni„n 2023. 

ECR menüsü

aws 
Amazon ECR Public Gallery 
u Reg 
Official 
Am 
(2.96 • downloads] 
by AWS 
Official AWS distributiM of Fluent Bit 
os/A„h: Linux, ARM 
eks-distro/kubernetes-csi/node-driver-registrar (2.88+ 
aws 
by Amazon EKS @ 
EKS Distro Kubemetes CSI Node Orim Registrar 
Linux, ARM 64 
eks-distro/kubernetes-csi/external-provisioner (2, 5B • 
by Amazon EKS @ 
Distro Kubemetes CSI 
Linus 06-64 ARM 64 
(1.70 
by 
provides stable, and 
applicatiwg. 
Linux, 06-64 ARM 64 
DATADOG 
downloads) 
'are 
eks•dist' 
by EXS Distribution 
EKS OiStm CSI 
ARM 64 
datadog/agent 12.7B' dcmnloads) 
by datadog @Vcifieg 
container for the new Oat*dcg Agent 
OS,'Arch: Limn, Windows, *86-64, ARM 64 
eks-distro/kubernetes/pause downloads) 
by EXS Distribution 
Amazon EKS OiStm Kubernetes pause image 
xg6-6a, ARM 
bitnami/aws-cli 
by @ 
AWS Command Line Interface (CLI) allows to AWS services 
'mm single it to multiple ærvicæ art actiMS 
through This listing is packaged by Bitnami, The 
in the offering are by 
use of them does irnply affiliation or endorsement, 
gitnami ire always to wwk right 
Wt Of the you enjoy kßing Bitnami for development. you Should 
VMmre Catalog needs 
https://tanwvmwaæ.cc•m/app•lication-catalog. Application Catalog 
a of cmtaineß Helm charts 
the Bitnami 
Os,'Arch: 06-64 

ECR ın public galerisi

Depolama: 
depolarda Çın depolama, GB'ay 0,10 USD 
Özel depolardan aktarılan veriler: 
Bölge: ABD D oğu h i o) 
İÇERİ Veri 
Tüm içeri aktarımımı 
DIŞARI Veri 
s„ki g,ggg 
dl) TB'ay 
s„ki 
Aylık TB'tan 
Genel depolardan aktarılan veriler: 
İÇERİ veri Aktarımı 
Tüm 'çeri 
Veri 
0,00 usa / GB 
GB 
na üSD 
na USD 
GB başı 
na USD 
GB 
na 0,05 uşD 
GB 0,00 USD 

5 TB ye kadar ücretsiz sonrasında cüzi bir ücretlendirme var.

$ docker run —it public.ecr.ans/amazonlinux/amazonlinux : Latest 

örneğin ecr dan amazonlinux image i çekme komutu.

docker run -it public.ecr.aws/amazonlinux/amazonlinux:latest bash

$ docker run 
oliuerpmydesktop: 
NME* " Amazon Linux" 
VERSION-"2023" 
Linux 2e23" 
—it public. ecr.ans/amazonlinux/amazontinux:latest bash 
: amazon: amazon_linux : 2023" 
HONE_URL="https : /laws.amazon.com/1inux/" 
3UG_REPORT _ URL; " https : / /github . 
SUPPORT 
bash-5.2# 

ECR Public Gall«y > bitnmi > bitnami/m«li 
usage 
bitnami/aws-cli 
downloads) 
by Bitnami 
u dap ago 
The AWS command Line Interface (CLO to AWS from a Sing 
is packaged by tndemarks in the offering are by t 
are mk right out Of the If eng" 
næds - Catalog prm/idé 
OS/Arch: xg6-6a 
Image tags 
AWS CLI packaged by Bitnami 

cli komutlarını çalıştırabileceğimiz bir image yapılmış.

public.ecr.aws/bitnami/aws-cli:latest

bu image in kullanılması için credentilas ların tanıtılması gerekir.

**docker run --rm --name aws-cli -v /home/oliver/.aws/config:/.aws/config -v /home/oliver/.aws/credentials:/.aws/credentials bitnami/aws-cli:latest s3 ls**

bu komut ile bilgisayardaki aws config dosyasını image e volume olarak bağlıyor.

env parametresiyle de access key ler tanıtılabilir, denemek lazım.

üstteki komutta image i çalıştırırken s3 ls komutunu da direk çalıştırdık.

şimdi konsolda kendimiz bir repo oluşturalım. Create repository dedik

ECR kullanılırken bi şirket public yapmaz, private repo için kullanır. Docker-hub da ücretli çünkü.

Repository name 
Provide a concise name. A developer should be able to identify the repository contents by the name. 
995714465210dkr.ecr.us-east-1amazonavwcom/ 

burada başındaki kısım hepimizin account idsi, sonra region kısmı. burası registry URL diye geçiyor.

ol'vercw / flask-app 
Last pushed: 2 8 
/ todopostgre 
Cmt&iny Last pushed: 8 
/ todonodejs 
Cmtgns: Last pushed: 3 8 

docker-hub ta image isimlerimiz bu şekildeydi.

image çekerken docker pull olivercw/flask-app:v2

image in ismi bu şekilde. ancak docker-hub ta bizim görmediğimiz başında docker.io/olivercw/flask-app:v2 olacak şekildedir ismi. burada docker-hub default registry olduğu için yazmıoruz.

ancak başka bir registry de isminin tam halini yazmamız gerekiyor.

995714465210.dkr.ecr.us-east-1.amazonaws.com/olivercw/flask-app:v2 şeklinde.

Visibility settings Info 
Choose the visibility setting for the repository. 
O Private 
Access is managed by IAM and repository policy permissions. 
O Public 
Publicly visible and accessible for image pulls. 
Repository name 
Provide a concise name. A developer should be able to identify the repository contents by the name. 
995714465210.dkr.ecr.us-east-l.amazonaws.com/ studentl-repo/todo-app 
22 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, 
hyphens, underscores, periods and forward slashes. 
Tag immutability Info 
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag 
immutability to allow image tags to be overwritten. 
Disabled 
@ Once a repository is created, the visibility setting of the repository can't be changed. 

repoya bu şekilde isim verelim. private seçili olacak en başta.

best practice olarak çalışan ekibin ismi sonra my-app olarak verilir.

teama-myapp şeklinde. image ler departmanlar için karışmasın, ayrışabilsin diye bu şekilde yapılır. her team app in başka kısmıyla ilgili çalışıyordur.

tag immutability tag değişsin mi değişmesin mi diye seçtiğimiz kısımdır.

tag immutability enable edersek aynı image den aynı tag ile iki kere gönderemeyiz. disable halindeyken aynı tag ile gönderirsek overwrite eder ve eskisini untagged olarak tutar.

Image scan settings 
Deprecation warning 
ScanOnPush configuration at the repository level is deprecated in favor of registry level scan filters. 
Scan on push 
Enable scan on push to ha'.æ each image automatically scanned after being pushed to a repository If disabled, each image scan must be 
manually started to get scan results. 
Disabled 

bu kısmı da image i push ederken scan etmesi (yani güvenlik taraması) içindir. güvenlik açığı olup olmadığına bakar. örneğin image içinde access secret keyler olmuş olabilir gibi.

burada uyarıri hatayı çözmez.

bu tür uyarıların ve çözümlerin olduğu bir database oluşturulmuş durumda, buradaki uyarıda o database e gönderiyor.

scan işlemi enable olursa push etme işlemi uzun sürer. scan işlemi başka bir yerde de yapılabiliyor.

Encryption settings 
KMS encryption 
You can use AWS Key Management Service (RMS) to encrypt images stored in this repository, instead of using the default encryption 
settings. 
Disabled 
@ The KMS encryption settings cannot be changed or disabled after the repository is created. 

burada da image i encrypte şekilde tutmak için bir seçenek mevcut.

create repository diyoruz ve repomuz oluştu.

docker da docker scan komutu var scan etmesi için.

Private 
Public 
Private repositories (1 of 1) 
View push commands 
Delete 
Q Find repositories 
Repository 
student 1- 
repo/todo-app 
995714465210.dkr.ecr.us-east- 
1.amazonaws.com/studentl-repo/todo- 
app 
Created at 
08 May'S 2023, 
(UTC+02) 
Tag 
immutability 
Disabled 
Actions A 
View 
Summary 
Images 
Permissions 
Lifecycle policies 
Repository tags 
Edit 
Repository 
Repository scan filters 
11 
rough 
ache 
ctive 

lifecycle policy lerler image lerin ne kadar süre tutulacağını belirliyoruz. çünkü bir yerden sonra şişme olabiliyor. birden fazla kural da belirtebiliyor ve bu kurallara öncelik verebiliyoruz.

Lifecycle rule configuration 
If apply this it be It s 
rules first. 
Rule priority 
rule in a 
Rule 
specify 
delete 

rule priority kutalın önceliği, kurala açıklama veriyoruz.

stat 
O untagged 
0 Any 
Match crit«ia 
specify is , 
an of in to 
• a limit 
Rule 
i, imagewiL rhiz 

bu şekliyle buradaki image larda son iki günü tut diğerlerini sil.

Image Status 
O Tagg«i 
O untagged 
Tag 
this is ifs. a list 
with 
Match Uit«ia 
specify is 
that of in '5 to 
it' limit on 
Rule 
is image„iu 

react olarak taglenenlerden son ikisi kalsın diğerlerini sil.

gün ve sayı olarak belirtebiliyoruz.

direk siler, arşive atmaz.

etkisi hemen görülmez, 24 saat içerisinde siler.

Create lifecycle rule 
Lifecycle rule configuration 
If you apply this lifæße plicy. it be It is that you dry your 
rules first. 
Rule p riwity 
specify a rule value do in a 
Rule 
s,mify 
status 
me is 
O Tagged 
untagged 
O Any 

yeni bir lifecycle oluşturmak istediğimizde rule priority yi direk 2 olarak getiriyor.

Edit JSON 
"delete , 
"tagStatW 
1 

lifecycle oluşturduktan sonra editlenebiliyor json formatında.

dokümantasyonda rule önceliği ile örnekler verilmiş:

Example A 
Repository contents; 
• Image A, Taglist: theta-I", "prod-I Pushed: 10 days ago 
• Image g, Taglist: [•beta-2•, •pa-2"1, Pushed: g days ago 
Image C, Taglist: C beta-gal, Pushed: g days ago 
Lifecycle policy text: 
"rulepriority" 
ription": "Rule 
"tagSt tu 
p • Ity"• 2 
"tagVrefixList": 
ription": "Rule 
"t aest tu 
"tagVrefixList": 
["prod"l. 

bu örnekleri incelemek için:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/lifecycle_policy_examples.html>

Private 
Public 
Private repositories (1 of 1) 
View push commands 
Delete 
Q Find repositories 
Repository 
name 
student 1- 
repo/todo-app 
Actions 
View 
Summary 
Images 
Permissions 
Create repo 
11 
ache 
995714465210.dkr.ecr.us-east- 
1.amazonaws.com/studentl-repo/todo- 
app 
Created at 
08 May'S 2023, 
(UTC+02) 
Tag 
immutability 
Disabled 
Lifecycle policies 
Repository tags 
Edit 
Repository 
Repository scan filters 

permissions ile de kullanıcıları repository içinfe sınırlama ve kişiye özel roller atama yapabiliyoruz.

şimdi ise image oluşturup gönderecez.

wget <https://github.com/awsdevopsteam/jenkins-first-project/raw/master/to-do-app-nodejs.tar>

komutuyla github tan to-do-app-nodejs.tar dosyasını çekelim.

• ec2-use 
to-do- app-nodejs. tar 
• ec2-use 
Is 
tar 
-xvf to-do-app-nodejs.tar 

tar komutuyla açtık.

to-do-app-node.js klasörü altında Dockerfile oluşturuyoruz.

FROM node:12-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "/app/src/index.js"]

dockerfile içeriğimiz bu şekilde.

yarn JavaScript in paket yöneticisi ve dependencyleri indiriyor burada.

docker build -t todo-app:latest .

docker build ile image i build edelim.

önce lokalimizde çalıştırmayı deneyip sonra ecr repoya gönderecez.

ecr repomuzda sağ üstte;

ECR > Repositwies Student 1 
studentl -repo/todo-app 
Imge u RI 
View pu 
Images (O) 
q m-tif«ts 
Image 
Deta [s 
V 
Arti type 
at 
V 
No 

push command ları gösteren bir buton var:

Push commands for studentl -repo/todo-app 
macOS 
Make you the of the AWS CLI and For Getting 
Started with A— ECR 
LIE the folming steps to authentiæte and push an image to additional registry 
methods, in&ding Amazon ECR aedential helper, see Registry Authentication 
1. an and uthentiæte yow LR.cker client to mr 
use the Aws 
aws get-login-pamord —regiM us-east-I login AWS --paßwod-stdin 
make of the AHS 
2. guild your image using the foll(m-ing information building a Oocker file scratch see the 
here an skip this step if is 
docker build .t studentl.repo/todo•app 
5. After the build completes, tag you can push the to this repository: 
046402772087 1 - 
4. Run the tollNing command to wsh this image to æted AWS rep&tory: 
dtRke-r push 046402772087 

docker image ls ile image i görelim:

• ec2-user@ip-172-31-95-121# to-do-app-nodejs:$ docker 
REPOSITORY 
todo-app 
TAG 
latest 
public . ecr. aws/ amazonlinux/ amazonlinux latest 
node 
12-al ine 
image Is 
IMAGE ID 
91e4694c871a 
595bc66dfd27 
bb6d28Ø39b8c 
CREATED 
About a minute ago 
6 days ago 
13 months ago 
SIZE 
179MB 
143MB 
91MB 
o ec2-user@ip-172-31-95-121# to-do-app-nodejs:$ 

container ayağa kaldırıp app i görelim:

docker run --name todo -d -p 80:3000 todo-app

browserda grelim:

ΛΛΙ1 't 
Νο •yet! Add 

docker rm -f todo ile container ı silelim.

şimdi image i ECR repoya gönderelim:

docker tag todo-app:latest 995714465210.dkr.ecr.us-east-1.amazonaws.com/student1-repo/todo-app:latest

komutuyla önce docker tag deyip image in ismini verdik sonra da vereceğimiz yeni ismi ve latest tag ini verdik.

eksi image i silip yerine yenisini yazmadı. şu anda image i yeni bir tag ile oluşturdu.

şimdi de docker push ile ECR repoya gönderelim:

@ ec2-user@ip-172-31-95-121# to-do-app-nodejs:$ docker push 995714465210. dkr.ecr.us-east-l. amazonaws. can/studentl-repo/ 
todo-app: latest 
The push refers to repository [995714465210. dkr. ecr. us-east-I. amazonaws . can/studentl-repo/todo-app] 
Idd4faecb93a : 
242f12216a79: 
667c99724da4: 
7f3Øcde3f699: 
fe810f59Ø2cc : 
dfd8cØ46c6Ø2 : 
4fc242d58285: 
no basic auth 
Preparing 
Preparing 
Preparing 
Preparing 
Preparing 
Preparing 
Preparing 
credentials 

no basic auth credentials diyor. yetkilendirme gerekiyor.

biz ECR a instance için full access yetkisi veriyoruz.

To authenticate Docker to an Amazon ECR private registry with the CLI 
To authenticate Docker to an Amazon ECR reglStry WIth get-login-password, run the awS ecr get—login-password command. When passing the authentication token to the docker login 
command, use the value AWS for the username and Specify the Amazon ECR registry URI want to authenticate to If authenticating to multiple registries, you must repeat the for 
each registry. 
A Important 
If you receive an error. install or upgrade to the latest version of the AVÆ CUI. For more information, see Installing the AWS Command Line Interface in the AWS Command Line Interface 
User Guide. 
get-login-password (AWS CLI) 
Get-ECRLoginCommand (AVIS Tools for Windows powerShell) 
login AWS 

docker ı amazon ecr private a cli ile authenticate etmek için bu komutu docs tan alıyoruz.

password al ve pipe ile bu bu password ü kullan ve bağlan demek bu komut.

bu komutu view push commands kısmından da alabiliyoruz:

Push commands for studentl-repo/todo-app 
x 
macOS / Linux 
Windows 
Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting 
Started with Amazon ECR Z. 
LJse the following steps to authenticate and push an image to your repository. For additional registry authentication 
methods, including the Amazon ECR credential helper, see Registry Authentication e. 
1. Retrieve an authentication token and authenticate your Docker client to your registry. 
Use the AWS CLI: 
aws ecr get-login-password —region us-east-I I docker login —username AWS —password-stdin 
995714465210.dkr.ecr.us-east-1.amazonaws.com 
Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed. 
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the 
instructions here e. You can skip this step if your image is already built: 
O docker build -t studentl-repo/todc—app . 
3. After the build completes, tag your image so you can push the image to this repository. 
docker tag studentl -repo/todo-app:latest 995714465210.dkr.ecr.us-east-l.amazonaws.com/studentl 
repo/todc—app:latest 
4. Run the following command to push this image to your newly created AWS repository: 
O docker push 
Close 

no Das1C autn 
creaentlals 
r@ip-172-31-95-121# to-do-app-nodejs:$ aws ecr get-login-password --region us-east-I I docker login 
• ec2-use 
AWS --password-stdin 995714465210. dkr. ecr. us-east-I. amazonaws . com 
WARNING! Your password will be stored unencrypted in /home/ec2-user/ . docker/config. json. 
Configure a credential helper to remove this warning. See 
https : / / docs . docker.com/engine/reference/commandline/login/#credentials-store 
- -username 
Login Succeeded 
to-do-app 
node I 

login oldu. .docker klasörü içinde config.json dosyası oluşturdu.

READMEmd config.jscn X 
"auths": 
"046492772087 .dkr.ecr.us-east-l.amazonaws .cœn": { 
"auth": 

config.json içindeki credentials kısmı aws ecr a docker image lerini alma yetkisi veriyor.

kubernetes cluster ında bunu secret ile kullanacağız bütün worker node larda çalışması için. capstone projede yapacağız bunu.

artık image i push edebiliriz.

• ec2-user@ip-172-31-95-121# to-do-app-nodejs:$ docker push 995714465210. dkr.ecr.us-east-l. amazonaws . can/studentl-repc 
todo-app: latest 
The push refers to repository [995714465210. dkr. ecr. us-east-I. amazonaws . can/studentl-repo/todo-app] 
Idd4faecb93a : 
242f12216a79: 
667c99724da4: 
7f3Øcde3f699: 
fe810f59Ø2cc : 
dfd8cØ46c6Ø2 : 
4fc242d58285: 
Pushed 
Pushed 
Pushed 
Pushed 
Pushed 
Pushed 
Pushed 
latest: digest: sha256:4ef5839f19e515288a7c1593fb5f28c4b3dda97b97ØbbØ6a7Ø832a92Ø13aa12d size: 
1787 

image ecr private repomuza gitti:

Amazon ECR > Repositories > studentl-repo/todo-app 
student I-repo/todo-app 
Images (1) 
Q Search artifacts 
View push commanl 
c 
Delete 
Scan 
status 
Details 
Vulnerabi 
Image tag 
latest 
Artifact type 
Image 
Pushed at 
08 Mays 
2023, 
(UTC+02) 
(MB) 
68.74 
Image 
Copy 
Digest 
sha256:4ef5839f19e5152... 

bunu tekrar latest diye tag lersek eskisini untagged yapacak ve yenisini latest olarak tagleyecek.

docker pull ile image i ecr repodan çekelim:

docker pull 995714465210.dkr.ecr.us-east-1.amazonaws.com/student1-repo/todo-app:latest

şimdi de docker run ile bu image i çalıştıralım. browser da gördük:

A Güvenli 54.83.149.171 
ment... Course: AWS 13-TU... Google Takvim 13... 
Your Repositories 
DeW3ps - Microsoft... 
(3) "devops" Searc.. 
HacklnS 
clem:e 
New Item 
Add Item 
No items yet! Add one above! 

container ı silelim.

docker rm -f todo komutuyla.

şimdi uygulamada bir değişiklik yapıyoruz.

EXPLORER 
v OPEN EDITORS 
Dockerfile 
G) READMEmd 
X JS app.js src/static"s 
v TO-DO-APP-NODEJS 
> spec 
v src 
> persistence 
> routes 
v static 
> css 
Dockerfile 
src > static > js > 
0 README-.md 
JS app.js 
JS app.js > TodoListCard 
46 
47 
48 
5e 
51 
52 
53 
54 
55 
56 
57 
58 
59 
PROBLEMS 
if 
C items] , 
(items 
null) return 
' Loading. ' ; 
return ( 
<React . Fragment> 
<AddItemForm onNewItem={onNewItem} 
{items . length 
<p >No items 
JS 
JS 
JS 
JS 
._app.Js 
. babel.min.js 
._react- bootstrap.js 
._react-dom.prod... 
OUTPUT 
{ items . map( item 
<ItemDisp1ay 
DEBUG CONSOLE TERMINAL 
PORTS 
7f3øcde3f699: Pushed 

56. satırında değişiklik yapıyoruz.

image i build edip tagliyoruz.

pushettik

student I-repo/todo-app 
Images (2) 
Q Search artifacts 
c 
Image tag 
V 
View push c 
Delete 
Scan 
status 
Artifact type 
Image 
Image 
Pushed at V 
08 Mays 
2023, 
(UTC+02) 
08 Mays 
2023, 
(UTC+02) 
(MB) 
68.74 
68.74 
Image 
Copy 
Copy 
Digest 
3 sha256:120405dedd9b8a.. 
0 sha256:4ef5839f19e5152. 

repo da ikinci push ettiğimiz image latest olarak geldi, çnceki untagged oldu.

docker run ile çalıştırdık.

Güvenli 54.83.149.171 
Course: AWS 13-TU... Google Takvim 13... 
Your Repositories 
DeW3ps - Microsoft... 
(3) "devops" Searc.. 
e; HacklnScience — s.. 
New Item 
Add Item 
No items yet Geovanni! Add one above! 

değişikliği gördük.

conatiner ı silelim.

şimdi ise lokalde çalışırken cli ile ecr da repo oluşturacaz.

bu komut sık kullanılan bir komut olacak bizim için.

aws ecr create-repository \

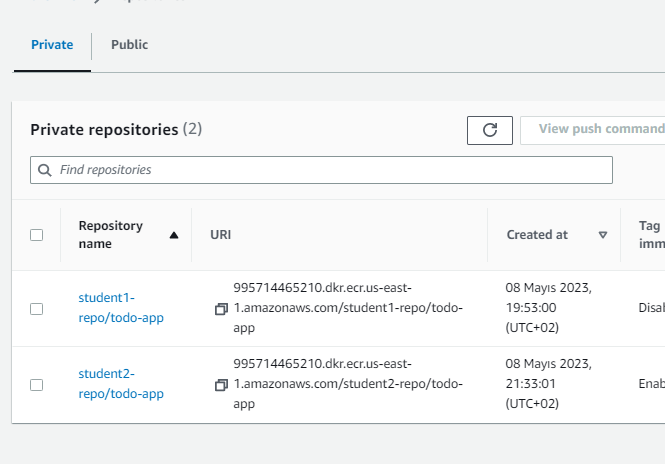
--repository-name student2-repo/todo-app \

--image-scanning-configuration scanOnPush=false \

--image-tag-mutability IMMUTABLE \

--region us-east-1

repo oluştu:

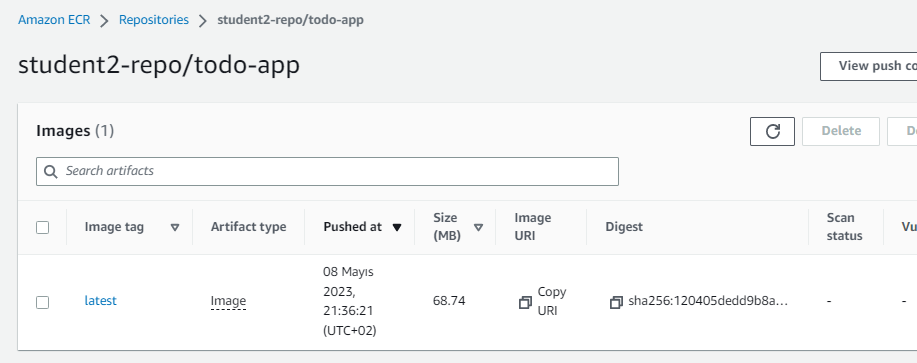


şimdi de tag leyip push ediyoruz:

docker tag todo-app:latest xxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/student2-repo/todo-app:latest

docker push xxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/student2-repo/todo-app:latest

push ladık ve image repoya gitti:



şimdi froce parametresiyle image leri silelim:

docker image rm --force <image\_id>

şimdi lokalde image yok ve ecr private repodan image i run edelim: direk run komutu girebiliriz.

docker run -dp 80:3000 995714465210.dkr.ecr.us-east-1.amazonaws.com/student2-repo/todo-app:latest

image i silmek için:

aws ecr batch-delete-image \

--repository-name student2-repo/todo-app \

--image-ids imageTag=latest

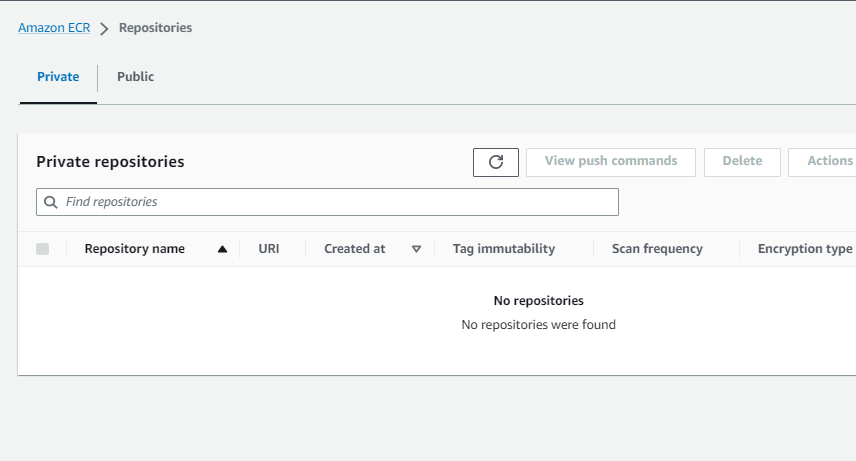
image silmeden repoyu silmek için:

aws ecr delete-repository \

--repository-name student2-repo/todo-app \

--force

ilk oluşturduğumuz repo içinde student2 yazan yerde 2 yi 1 yapıyoruz.



repolarımız silindi.